

**ANO
2025**



UNINTER

PROJETO MULTIDISCIPLINAR

ORIENTAÇÕES E ESTUDO DE CASO

Prof. Winston Sen Lun Fung, Me.

INTRODUÇÃO

Olá a todos.

Sejam todos muito bem-vindos!

Nesta atividade final de Projetos, você terá a oportunidade de integrar os conhecimentos adquiridos ao longo do curso e aplicá-los em um Estudo de Caso na área de saúde. O objetivo é que você desenvolva um **sistema teórico** (com protótipo, documentação ou código funcional) que atenda aos requisitos propostos, levando em consideração os princípios do desenvolvimento de software e da aplicação da engenharia de Software com ênfase correspondente à sua rota de formação escolhida (Back-end, Front-end ou Qualidade de Software).

Este estudo de caso foi elaborado para simular um cenário real de desenvolvimento de um sistema de alta criticidade, onde a segurança dos dados, a qualidade do software e a usabilidade são fundamentais. Aproveite esta oportunidade para demonstrar as competências adquiridas, valorizar seu portfólio e aprimorar seu aprendizado de forma prática.

Desejamos um excelente trabalho e estamos à disposição nos canais de tutoria para esclarecimento de dúvidas!

*No mais, desejamos bom desenvolvimento da atividade prática em nome dos professores
da disciplina de Projeto Multidisciplinar.*



SUMÁRIO

INTRODUÇÃO	1
ESTUDO DE CASO: SISTEMA DE GESTÃO HOSPITALAR E DE SERVIÇOS DE SAÚDE (SGHSS)	3
ORIENTAÇÕES GERAIS	4
ESTRUTURA DA ATIVIDADE	5
CRITÉRIOS DE AVALIAÇÃO	7
DICAS E BOAS PRÁTICAS	7
ENTREGA	7



ESTUDO DE CASO: SISTEMA DE GESTÃO HOSPITALAR E DE SERVIÇOS DE SAÚDE (SGHSS)

A instituição **VidaPlus** administra hospitais, clínicas de bairro, laboratórios e equipes de home care. Ela precisa de um **Sistema de Gestão Hospitalar e de Serviços de Saúde (SGHSS)** para centralizar:

1. **Cadastro e Atendimento de Pacientes:** consultas, exames, prontuários, telemedicina.
2. **Gestão de Profissionais de Saúde:** médicos, enfermeiros, técnicos, agendas, prescrições.
3. **Administração Hospitalar:** leitos, relatórios financeiros, suprimentos.
4. **Telemedicina:** atendimentos e prescrições online, marcação de consultas presenciais e exames.
5. **Segurança e Compliance:** controle de acesso, LGPD, registros de auditoria.

Requisitos Funcionais e Não Funcionais

- **Pacientes:** cadastrar dados, visualizar histórico clínico, agendar/cancelar consultas, receber notificações, acessar teleconsulta.
- **Profissionais de Saúde:** gerenciar agendas, atualizar prontuários, emitir receitas digitais, acompanhar histórico dos pacientes.
- **Administradores:** gerenciar cadastros (pacientes, profissionais), controlar fluxo de internações, gerar relatórios.
- **Telemedicina:** realizar videochamadas seguras, registrar prontuários e prescrições online.
- **Segurança:** criptografia de dados sensíveis, controle de acesso por perfil, registro de logs e auditoria, conformidade com a LGPD.
- **Escalabilidade:** suportar múltiplas unidades hospitalares;
- **Desempenho:** tempo de resposta rápido em consultas críticas;
- **Acessibilidade:** interface amigável e responsiva, com padrões W3C/WCAG;
- **Disponibilidade:** no mínimo 99,5%, com backups e logs robustos.

ORIENTAÇÕES GERAIS

1. Leitura do Material

- a. Consulte o conteúdo teórico da disciplina e revise conceitos de Engenharia de Software, Modelagem, Desenvolvimento de Sistemas e Qualidade.
- b. Caso surjam dúvidas, use os canais de tutoria ou fóruns de discussão disponíveis no Ambiente Virtual de Aprendizagem (AVA).

2. Consultas e Pesquisas

- a. Você pode (e deve) pesquisar livros, artigos, repositórios e outras fontes de conhecimento.
- b. Lembre-se de referenciar corretamente as fontes utilizadas.

3. Evite Plágio

- a. Produza suas próprias telas, diagramas, scripts e textos. Cada projeto de TI costuma ter suas características e variáveis personalizadas.
- b. Trabalhos idênticos ou cópias literais da internet serão tratados como plágio e podem resultar em nota zero.

4. Evite “dividir” o projeto em partes isolada

- a. Procure integração entre as partes (back-end, front-end, testes).

5. Repositório

- a. **Se a sua ênfase for Back-end:** **indique o endereço do repositório com os códigos fonte.** (GitHub, Git Lab, Bitbucket etc).
- b. **Se a sua ênfase for Front-end:** **indique o endereço com os protótipos das telas.** (figma, canvas etc). Em caso de telas codificadas (html, css, Javascript e frameworks) **indicar o endereço de publicação das páginas** (https:// ...) e; **indicar e utilizar** repositórios como GitHub, Git Lab, Bitbucket etc.
- c. **Se a sua ênfase for Qualidade de Software:** **quando necessário** repositórios como GitHub, Git Lab, Bitbucket entre outros para armazenar planos de testes, e resultados.

ATENÇÃO: VERIFIQUE OS DIREITOS DE ACESSO AOS LINKS, PROTOTIPOS E REPOSITÓRIOS. A responsabilidade de disponibilização dos acessos é do Aluno(a), caso não esteja disponível o acesso na correção esse item receberá nota ZERO.



ESTRUTURA DA ATIVIDADE

Documento Principal

Você deverá produzir **um arquivo único em formato PDF** seguindo a estrutura mínima:

1. Capa e Sumário

- Identifique o curso, a disciplina, seu nome e seu RU, polo de apoio, semestre e professor.
- Faça um sumário com a organização do documento.

2. Introdução

- Apresente o contexto do estudo de caso, definindo os objetivos do projeto, os principais usuários e a relevância do sistema.

3. Análise e Requisitos

- Descreva os requisitos funcionais e não funcionais (podem ser detalhados em tabelas ou listagens).
- Se necessário, inclua um Diagrama de Casos de Uso ou outro artefato UML para ilustrar os processos.

4. Modelagem e Arquitetura

- **Se a sua ênfase for Back-end:** inclua diagrama de classes, DER (diagrama entidade-relacionamento), descrição dos principais endpoints da API, tecnologias de persistência etc.
- **Se a sua ênfase for Front-end:** apresente wireframes, protótipos de tela, design responsivo, frameworks escolhidos etc.
- **Se a sua ênfase for Qualidade de Software:** descreva a estratégia de testes (funcionais, não funcionais, de segurança, automação), planos de teste (listar), possíveis ferramentas (Selenium, JMeter, OWASP ZAP etc.).

5. Implementação (Prototipagem)

- Desenvolva um protótipo ou uma versão funcional mínima do sistema.
- Caso não seja possível o desenvolvimento completo, apresente **pseudocódigo e exemplos** de como seria a implementação.
- Faça a indicação do link/endereço repositório (git) do projeto desenvolvido. Ou, indique o link/endereço para o projeto de prototipagem (figma, cavaas etc.).

6. Plano de Testes

- Descreva casos de teste, critérios de aceitação, possíveis roteiros para testes de carga, segurança e usabilidade.



- **Se a sua ênfase for Back-end:** fazer um resumo.
- **Se a sua ênfase for Front-end:** fazer um resumo.
- **Se a sua ênfase for Qualidade de Software:** descreva os planos de testes (funcionais, não funcionais, de segurança, automação), possíveis ferramentas (Selenium, JMeter, OWASP ZAP etc.).

7. Conclusão

- Aborde as principais **lições aprendidas**, desafios e pontos de atenção para evoluções futuras do projeto.

8. Referências

- Liste livros, sites, artigos e quaisquer outras fontes que subsidiaram seu trabalho.

Materiais Suplementares (Anexos)

- **Modelos UML** (diagramas de classes, diagramas de atividade, diagrama de sequência, diagramas de estados, diagramas de componentes, diagrama de implantação se julgarem necessário).
- **Código fonte importantes** podem ser disponibilizados para apoiar indicações no texto. Indicar o endereço do repositório onde eles estão armazenados.
- **Prints de Tela** ou screenshots de protótipos e testes.
- **Scripts de Teste** ou resultados de ferramentas de automação (caso tenha realizado).

Observação: Não se esqueça de converter tudo em **PDF** único antes de postar no AVA.

CRITÉRIOS DE AVALIAÇÃO

1. **Estrutura e Organização (10%)**
 - a. Clareza na apresentação, diagramação e coerência textual.
2. **Qualidade da Documentação (30%)**
 - a. Nível de detalhamento dos requisitos, diagramas e explicações técnicas.
3. **Modelagem/Arquitetura ou Protótipo (30%)**
 - a. Precisão e consistência dos diagramas (ou protótipos/telas) com os requisitos propostos.
 - b. Indicação do link/endereço repositório com os códigos ou link/endereço para ferramenta de prototipagem.
4. **Plano de Testes e Estratégia de Qualidade (20%)**
 - a. Definição de testes funcionais, não funcionais, segurança e automação.
5. **Originalidade e Aplicação Prática (10%)**
 - a. Aderência ao cenário, soluções criativas, referências a boas práticas do mercado.

DICAS E BOAS PRÁTICAS

1. **Cronograma:** divida a atividade em etapas (requisitos, modelagem, implementação, testes) para não deixar tudo para o último momento.
2. **Ferramentas Úteis:**
 - a. Modelagem: Lucidchart, Draw.io, Astah, Visual Paradigm.
 - b. Protótipo de Telas: Figma, Adobe XD, Canvas, Marvel App.
 - c. Testes: Selenium, Cypress, JMeter, OWASP ZAP.
 - d. Documentação: Google Docs, Microsoft Word, Latex.
3. **Revisão Final:** antes de postar, revise o PDF para garantir que todos os itens solicitados estejam presentes.

ENTREGA

1. **Formato:** Um arquivo único em **PDF** (nomeado como "Projeto_RU_NomeDoAluno.pdf").
2. **Local de Envio:** Área de "Trabalhos" do AVA-Univirtus.
3. **Data de Entrega:** Verifique o Calendário Acadêmico no AVA.
4. **Verifique se todos os links e endereços possuem direitos de acesso aos corretores e se estão funcionais.**